

## PACKET ORDER DETERMINING METHOD AND APPARATUS

### Field of the Invention

5       The present invention relates generally to packet switched networks and, in particular, to variable delay packet switched networks.

### Background of the Invention

10       Some communications systems utilize packet switched networks that allow packets of information to arrive at a receiver in a different order than they were transmitted. Some applications are sensitive to the received order of packets, and it can be desirable to be able to identify when packets are received out of order so that corrective steps can be taken.

15       Secure networks are one example of an application sensitive to the received order of packets. In some secure networks, encryption keys used at the transmitter and receiver are synchronized. When encrypted packets are received out of order at the receiver, synchronization can be lost. Another example of an application sensitive to the received order of packets is voice data as voice data processed out of order at a receiver can result in degraded voice quality output.

20       In order to ensure proper ordering of packets, portions of packets can be dedicated to hold packet-ordering information. For example, a field in a packet header can be used to hold information specifying the relative order of a packet. When the packet contents are read, packets received out of order can be re-arranged into the correct order. However, utilizing portions of a packet to specify relative packet order consumes communications channel bandwidth. It would be desirable for a communications system to be able to specify the order of transmitted packets without consuming communications channel bandwidth.

25       Accordingly, a significant need exists for a method and apparatus for communicating packet ordering in communications systems without consuming communications channel bandwidth.

Brief Description of the Drawings

The invention is pointed out with particularity in the appended claims. However, a more complete understanding of the present invention may be  
5 derived by referring to the detailed description and claims when considered in connection with the figures, wherein like reference numbers refer to similar items throughout the figures, and wherein:

FIG. 1 shows a packet switched communications system in accordance with a preferred embodiment of the present invention;

10 FIG. 2 shows variable latencies in a packet switched communications system in accordance with a preferred embodiment of the present invention;

FIG. 3 shows relative ordering of packets in a packet switched communications system in accordance with an embodiment of the present invention;

15 FIG. 4 shows a transmit processing block in accordance with a preferred embodiment of the present invention;

FIG. 5 shows a receive processing block in accordance with a preferred embodiment of the present invention;

20 FIG. 6 shows a mask store in accordance with a preferred embodiment of the present invention;

FIG. 7 shows a transmit processing block in accordance with an alternate embodiment of the present invention;

FIG. 8 shows a receive processing block in accordance with an alternate embodiment of the present invention;

25 FIG. 9 is a flowchart of a method for adding packet ordering information to a plurality of data packets in accordance with a preferred embodiment of the present invention;

30 FIG. 10 is a flowchart of a method for determining the transmitted order of a received packet relative to other received packets in accordance with a preferred embodiment of the present invention;

FIG. 11 is a flowchart of a method for determining the transmitted order of a received packet relative to other received packets in accordance with an alternate embodiment of the present invention; and

35 FIG. 12 is a flowchart of a method for determining a packet order of a received packet in accordance with a preferred embodiment of the present invention.

# Detailed Description of the Drawings

In the following detailed description, reference is made to the accompanying drawings that show, by way of illustration, specific  
5 embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

The method and apparatus of the present invention provide a mechanism for determining the relative transmission order of packets in a  
10 packet switched network. Transmitted packets have error detection or forward error correction applied, and then they are masked with "ordering masks" prior to transmission. The ordering masks are applied to packets in an order that is known to both the transmitter and receiver. The receiver unmask  
15 s received packets and performs error detection. The proper relative transmission order of the packet is found when the correct ordering mask is used, resulting an acceptable error rate.

Turning now to the drawings in which like reference characters indicate corresponding elements throughout the several views, attention is first directed to FIG. 1. FIG. 1 shows a packet switched communications system  
20 in accordance with a preferred embodiment of the present invention. Communications system 100 includes communications devices 101 and 111 that communicate over channel 130. Communications system 100 can include any number of communications devices.

Communications device 101 includes voice processing block 104,  
25 transmit processing block 110, data formatting block 114, and packet transmitter 120. Voice processing block 104 receives voice signals on node 102, processes them, produces packets representing the voice signals, and send the packets to transmit processing block 110. Voice processing block 104 can include a microphone, an audio circuit for conditioning and sampling,  
30 and a vocoder. Data formatting block 114 receives data on node 112 and formats it prior to transmission. Formatting can include inserting data in packets and adding packet header information.

Voice processing block 104 and data formatting block 114 are examples of packet formatting blocks. The packet formatting blocks output  
35 packets that are input to transmit processing block 110. Other types of packet formatting blocks can be used without departing from the scope of the present invention. For example, transmit processing block 110 can receive

formatted packets from a separate data processing system or a separate communications system. In these embodiments, the separate data processing system and separate communications system are examples of packet formatting blocks.

5 Transmit processing block 110 receives voice packets on node 106 and data packets on node 116. Transmit processing block 110 applies ordering masks to the packets such that the transmitted order of the packets can be discerned at the receiver, even when the packets are received out of order at the receiver. Embodiments of transmit processing block 110 are  
10 discussed below with reference to FIGS. 4 and 7.

Transmit processing block 110 sends packets to packet transmitter 120 on node 118. Packet transmitter 120 performs additional processing prior to transmission, such as interleaving and modulation. Packet transmitter 120 transmits blocks of data into channel 130.

15 Channel 130 can be any communications channel that allows packets to arrive at a receiver in a different order than they were transmitted. For example, channel 130 can be a satellite communications system that provides multiple communications paths, or it can be a network in which packets can take different routes to a destination, such as the Internet.  
20 Multiple communications paths are shown in channel 130 as paths 132 and 134. If both paths 132 and 134 are utilized by communications system 100, and paths 132 and 134 exhibit different end-to-end delay, packets can arrive at packet receiver 138 in a different order than they were transmitted by packet transmitter 120.

25 Communications device 111 includes packet receiver 138, receive processing block 142, voice processing block 146, and data formatting block 152. The blocks in communications device 111 generally perform the reverse of the corresponding blocks in communications device 101. For example, packet receiver 138 demodulates, de-interleaves, and sends packets to  
30 receive processing block 142 on node 140. Voice packets are sent from receive processing block 142 to voice processing block 146 on node 144, and data packets are sent from receive processing block 142 to data formatting block 152 on node 150. Voice processing block 146 performs the inverse operation of voice processing block 104 and outputs voice signals on node  
35 148, and data formatting block 152 performs the inverse operation of data formatting block 114 and outputs data signals on node 154.

FIG. 2 shows variable latencies in a packet switched communications system in accordance with a preferred embodiment of the present invention. Latencies 210 and 220 represent the minimum and maximum latencies in a network. For example, latency 210 can represent the end-to-end delay of path 132 in channel 130 (FIG. 1) when path 132 is the shortest available path, and latency 220 can represent the end-to-end delay of path 134 when path 134 is the longest available path.

*sub a1* → ~~The difference between the shortest and longest delay yields the~~  
"maximum latency variability." The maximum latency variability measures the largest difference in transmission time between two packets that can be expected to arrive out of order. For example, if the minimum latency of a network is 100 milliseconds, and the maximum latency of the network is 400 milliseconds, then the maximum latency variability of the network is 300 milliseconds. Packets transmitted less than the maximum latency variability apart in time, or less than 300 milliseconds apart, may arrive at a receiver out of order. In contrast, packets transmitted greater than 300 milliseconds apart cannot arrive out of order because they are transmitted greater than the maximum latency variability apart in time. Minimum latency 210 is shown as a finite latency that is a substantial portion of maximum latency 220. In some embodiments, minimum latency 210 is very small, and it approaches zero, as ~~is the case when a transmitter and receiver have a very direct path.~~

FIG. 3 shows relative ordering of packets in a packet switched communications system in accordance with an embodiment of the present invention. Transmitted packets 302, 304, and 306 are transmitted into channel 310. They are received from channel 310 as packets 316, 318, and 320. The order of transmission is shown by direction 308. First packet A is transmitted, followed by packet B, and then followed by packet C. The order of reception is shown by direction 322. First packet B is received, followed by packet C, and then followed by packet A. Packet A has been received out of order by two packets.

Channel 310 is shown with two paths: path 312 and path 314. The out-of-order reception of packets can be caused by packets taking different paths. For example, if the latency of path 312 is greater than the latency of path 314 by two packet transmission times, and packet A is transmitted on path 312, the out-of-order reception shown in FIG. 3 results.

In some embodiments of the present invention, packets received out of order are discarded when they are older than previously received packets.

For example, packet A would be discarded because it was received out of order and it is older than packets A and B, which have already been received. Packets B and C, although received out of order, would not be discarded, because they are not older than previously received packets. These  
5 embodiments can be advantageous in applications such as voice processing. When coded speech is being transmitted, sending packets to the vocoder out of order can cause undesirable results. When a packet is discarded, modern vocoders estimate the missing speech and produce good quality results.

In other embodiments of the present invention, packets received out of  
10 order are buffered, and the correct order is reconstructed as other out-of-order packets arrive. In these embodiments, in the example of FIG. 3, packets B and C are buffered, and packet A is restored to its proper location prior to sending the packets on. These embodiments can be advantageous in applications such as encrypted communications and data communications.  
15 In encrypted communications, packet order can be important in part for maintaining sequences of cryptographic keys between transmitters and receivers. In data communications, packet order can be important, in part because out-of-order packets can cause corrupt files.

FIG. 4 shows a transmit processing block in accordance with a  
20 preferred embodiment of the present invention. Transmit processing block 400 is an embodiment of transmit processing block 110 (FIG. 1). Transmit processing block 400 includes encryptor 406, forward error device 416, masking device 420, mask store 424, and controller 412.

Voice packets are received on node 402 which corresponds to node  
25 106 in FIG. 1, and data packets are received on node 404 which corresponds to node 116 in FIG. 1. For ease of description, nodes 402 and 404 are both shown entering encryptor 406. One skilled in the art will understand that a multiplexor or other suitable circuit can be utilized to multiplex the voice and data packets into encryptor 406.

*sub 30*  
*a2* → ~~Encryptor 406 encrypts packets and sends encrypted packets to~~  
forward error device 416. Encryptor 406 receives encryption keys on node 408. Node 408 can be coupled to a key generator that is internal or external to communications device 101 (FIG. 1). In a preferred embodiment, encryptor 406 is an exclusive-or (XOR) operator that XOR's the packets with  
35 the encryption keys. In other embodiments, encryptor 406 utilizes other operators, such as public key cryptography.

Forward error device 416 performs operations such as applying error detection codes, applying error correction codes, and performing compression. Typical error detection codes such as cyclical redundancy check (CRC), and forward error codes (FEC) are well known in the art, and they need not be explained further here.

In some embodiments, forward error device 416 applies error codes differently as a function of the data source. For example, when voice packets are received on node 402, the operation of forward error device 416 can be different than when data packets are received on node 404, in part because it can be desirable to apply more coding to some portions of speech than to others.

Packets having error codes applied are sent to masking device 420 on node 418. Masking device 420 applies ordering masks to the packets to produce masked packets on node 426. Ordering masks are received from mask store 424 on node 422. In a preferred embodiment, ordering masks are digital words having the same size as the packets received on node 418. However, it should be understood that ordering masks could be larger or smaller than the packets received on node 418. Ordering masks are applied to packets using an operation that can be performed in reverse at a receiver. In a preferred embodiment, ordering masks are applied to packets using an XOR operation.

Mask store 424 includes a list of ordering masks in a known order. Each ordering mask is unique, and each ordering mask is applied to one packet at a time, and in order. In some embodiments, mask store 424 is a circular buffer and each ordering mask is used repeatedly or at least more than once. In a preferred embodiment, mask store 424 includes a number of masks sufficient to allow a unique ordering mask to be applied to each packet transmitted within a time period equal to the maximum latency variability. Mask store 424 is discussed further with reference to FIG. 6 below.

Controller 412 is coupled to encryptor 406, forward error device 416, masking device 420, and mask store 424, by bus 410. In some embodiments, controller 412 is a processor such as a microprocessor, digital signal processor, or the like. In other embodiments, controller 412 is dedicated hardware, such as a state machine. Controller 412 operates to maintain proper timing between the various devices.

In some embodiments, encryptor 406 is omitted, and packets are received by forward error device 416 from a packet formatter such as voice

processing block 104 or data formatting block 114. Packets undergo the application of error detection codes or forward error correction codes, and then undergo masking prior to transmission.

5 FIG. 5 shows a receive processing block in accordance with a preferred embodiment of the present invention. Receive processing block 500 is an embodiment of receive processing block 142 (FIG. 1). Receive processing block 500 includes unmasking device 504, mask store 506, error detection device and buffer 514, decryptor 520, and controller 516.

10 Receive processing block 500 receives packets on node 502 from packet receiver 138 (FIG. 1). Packets on node 502 have been encrypted, have had error coding applied, and have been masked by transmit processing block 400. As discussed above, packets arriving on node 502 may be out-of-order packets, such as those shown in FIG. 3. Receive processing block 500 determines whether the packets on node 502 are out-  
15 of-order. In some embodiments, out-of-order packets are discarded, and in other embodiments, out-of-order packets are buffered and the correct order is restored.

Unmasking device 504 receives masked packets on node 502, and it applies ordering masks from mask store 506 to the masked packets. In a  
20 preferred embodiment, unmasking device 504 is an XOR operator that receives ordering masks on node 508 and XOR's the ordering masks with received packets on node 502. When a received packet is in the correct order, the ordering mask received from mask store 506 on node 508 will correctly unmask the received packet. Unmasking device 504 sends the  
25 unmasked packet on node 512 to error detection device and buffer 514. Error detection device and buffer 514 performs the reverse of forward error device 416. When the correct ordering mask has been used, as is the case in this example, no errors are found.

When a received packet is out-of-order, the first ordering mask  
30 received from mask store 506 will not correctly unmask the received packet. As a result, error detection device and buffer 514 will detect errors. When errors are detected by error detection device and buffer 514, different ordering masks from mask store 506 are sent to unmasking device 504. Different ordering masks are used to unmask the received packets until error  
35 detection device and buffer 514 reports an acceptable amount of errors. Because the order of ordering masks in mask store 506 is known to correspond to the transmitted order of packets, the correct transmitted order



of received packets can be discerned at the receiver, even when received out of order.

Sub  
a3

~~In one embodiment, error detection device and buffer 514 buffers~~  
packets that are received out of order. When enough packets have been  
5 received to re-create the correct transmitted order, error detection device and  
buffer 514 outputs packets in the correct order on node 518. The re-creation  
of the correct transmitted order is advantageous in part because decryptor  
520 receives encryption keys that are synchronized with encryption keys  
used at the transmitter. When packets are received in order at decryptor 520,  
10 synchronization is maintained. In contrast, when packets are not received in  
~~order at decryptor 520, synchronization can be lost.~~

In another embodiment, error detection device and buffer 514 discards  
out-of-order packets that are older than packets previously received. In this  
embodiment, error detection device and buffer 514 may not include a buffer.  
15 Instead, error detection device and buffer 514 sends packets that are not  
discarded on node 518 to decryptor 520, and decryptor 520 receives packet  
order information either from error detection device and buffer 514 on node  
518, or from controller 516 on bus 510. In this manner, decryptor 520 can  
maintain synchronization even though packets have been discarded. This  
20 embodiment can be advantageous when receive processing block 500 is  
used in secure voice applications. Some packets can be discarded without  
materially degrading a voice signal. By allowing packets to be discarded,  
processing complexity can be reduced, and data throughput can be  
increased.

Controller 516 communicates with other devices in receive processing  
block 500 using bus 510. In a preferred embodiment, controller 516 keeps  
track of the correct order of ordering masks in mask store 506. Controller 516  
also receives error information from error detection device and buffer 514. In  
this manner, controller 516 can step through mask store 506 and cause  
30 unmasking device 504 to apply ordering masks until an acceptable amount of  
errors are reported by error detection device and buffer 514. In some  
embodiments, controller 516 is a processor such as a microprocessor, a  
digital signal processor, or the like. In other embodiments, controller 516 is  
made up of dedicated hardware, such as a state machine. In some of these  
35 embodiments, controller 516 is distributed among the other devices such that  
each device provides control circuitry sufficient to practice the present  
invention.

sub  
a4

FIG. 6 shows a mask store in accordance with a preferred embodiment of the present invention. Mask store 600 includes a list of ordering masks 610. Also shown in FIG. 6 is ordering mask pointer 620. Ordering mask pointer 620 can point to a current ordering mask. When mask store 600 is utilized in a transmit processing block, such as mask store 424, the current ordering mask is the ordering mask corresponding to the most recently transmitted packet. When mask store 600 is utilized in a receive processing block, such as mask store 506, the current ordering mask is the ordering mask corresponding to the most recently received packet. As packets are transmitted and received, current ordering mask pointer 620 is advanced in the list of ordering masks within mask store 600. For example, mask A can be applied to a first packet, mask B can be applied to a second packet, and mask C can be applied to a third packet. After mask C has been applied, current ordering mask pointer 620 points to mask C as shown in FIG. 6.

When packets are received out of order as shown in FIG. 3, current ordering mask pointer 620 can point to mask C prior to receiving packet A which corresponds to mask A. When the next packet is received, the receive processing block will first use mask D. The error detection device will report errors because the correct ordering mask is mask A. Ordering masks from mask store 600 will be used in order through the last mask. In an embodiment that discards out-of-order packets that are older than previously received packets, the received packet will be thrown away. In an embodiment that re-creates the transmitted order of out-of-order packets, the remaining masks will be applied. Ordering mask A will be found to be the correct ordering mask for the received packet, and the packet will be placed in the correct order in the received data packet stream. Mask store 600 preferably maintains enough masks to discern the correct transmitted order of any packets transmitted within the maximum latency variability of each other.

sub  
a5

FIG. 7 shows a transmit processing block in accordance with an alternate embodiment of the present invention. Transmit processing block 700 includes forward error device 706, encryptor/masking device 716, and controller 710. Transmit processing block 700 does not encrypt packets prior to forward error device 706. Forward error device 706 receives voice packets on node 702 and data packets on node 704, and it applies forward error codes thereto. Packets having error codes applied are sent to encryptor/masking device 716 on node 712. In one embodiment, encryptor/masking device 716 encrypts packets and masks them in a single

operation. Encryption keys are received on node 714, and they are used to encrypt packets received on node 712. The encryption keys received on node 714 are also used to mask the packets. Because the order of encryption keys is known at both the transmitter and the receiver, encryption keys can be used as ordering masks.

Controller 710 is coupled to forward error device 706 and encryptor/masking device 716 by bus 708. Controller 710 can be a processor such as a microprocessor, digital signal processor, or the like. Controller 710 can also be dedicated hardware, such as a state machine. In addition, controller 710 can be distributed hardware such that forward error device 706 and encryptor/masking device 716 provide their own control circuitry.

FIG. 8 shows a receive processing block in accordance with an alternate embodiment of the present invention. Receive processing block 800 includes decryptor/unmasking device 804, error detection device and buffer 812, and controller 814. The operation of receive processing block 800 is analogous to the operation of receive processing block 500 (FIG. 5). Packets received on node 802 are unmasked by decryptor/unmasking device 804. Unmasked packets are sent to error detection device and buffer 812 on node 810. Error detection device and buffer 812 performs error detection and reports errors to controller 814 on bus 808. When errors are found, an incorrect mask was used in the unmasking operation by decryptor/unmasking device 804.

Decryptor/unmasking device 804 utilizes encryption keys received on node 806 as ordering masks to unmask packets received on node 802. When the correct encryption key is used to unmask a received packet, the packet is unmasked and decrypted in a single operation.

Error detection device and buffer 812 operates in a fashion analogous to error detection device and buffer 514 (FIG. 5). In some embodiments, error detection device and buffer 812 buffers packets received out of order and re-creates the correct order. In other embodiments, error detection device and buffer 812 discards packets that are older than previously received packets. Error detection device and buffer 812 outputs voice data on node 816 and non-voice data on node 818.

Controller 814 communicates with decryptor/unmasking device 804 and error detection device and buffer 812 utilizing bus 808. Controller 814 can be a processor such as a microprocessor, digital signal processor, or the

like. Controller 814 can also be dedicated hardware such as a state machine, either centralized or distributed among the various devices.

FIG. 9 is a flowchart of a method for adding packet ordering information to a plurality of data packets in accordance with a preferred embodiment of the present invention. Method 900 begins in block 910 when error detection is applied to each of a plurality of data packets. Block 910 corresponds to the operation of forward error device 416 (FIG. 4), or forward error device 706 (FIG. 7). In block 920, each of the plurality of data packets is masked using ordering masks. The ordering masks are maintained in a known order, such as those shown in FIG. 6. The ordering masks and the known order are also known by a receiver such that the receiver can discern the relative packet order of each of the plurality of data packets.

FIG. 10 is a flowchart of a method for determining the transmitted order of a received packet relative to other received packets in accordance with a preferred embodiment of the present invention. Method 1000 begins in block 1010 when a packet is received. As described in block 1020, blocks 1030, 1040, and 1050 are successively traversed for each ordering mask in a list of ordering masks. In block 1030, an ordering mask is used to unmask the received packet. In block 1040, the unmasked received packet is checked for errors. In block 1050, the errors in the unmasked received packet are compared to a threshold. If the errors are below a threshold, processing proceeds with block 1060. If errors are above a threshold, another ordering mask is utilized beginning with block 1020.

If, in block 1020, all ordering masks have been successively tried without finding the correct ordering mask, as shown in block 1080, the received packet is discarded in block 1090.

When the correct ordering mask is found, ordering masks older than the correct ordering mask are removed from the list of ordering masks in block 1060. In block 1070, the unmasked received packet is decrypted.

Method 1000 corresponds to the operation of a receive processing block that discards received packets that are older than previously received packets. For example, when method 1000 is applied using receive processing block 500 (FIG. 5), ordering masks are removed from mask store 506 when they correspond to packets that are older than previously received packets. When the remaining ordering masks in mask store 506 are applied, and the correct ordering mask is not found, the received packet is discarded.

FIG. 11 is a flowchart of a method for determining the transmitted order of a received packet relative to other received packets in accordance with an alternate embodiment of the present invention. Method 1100 begins in block 1110 when a temporary ordering mask is set equal to a next ordering mask in a list of ordering masks. The ordering masks are maintained in order of packet transmission, such as the list of ordering masks shown in FIG. 6. The temporary ordering mask can be located in a register maintained within unmasking device 504 or mask store 506 (FIG. 5), for the purpose of performing an unmasking operation. The temporary ordering mask can be modified without physically modifying the ordering masks within the mask store.

In block 1120, the temporary ordering mask is applied to a received packet to produce an unmasked received packet. The unmasked received packet corresponds to a packet on node 512. In block 1130, the unmasked received packet is checked for errors. If errors are found in block 1140, a next ordering mask is tried again in block 1110. When errors are not found, a current ordering mask is set equal to the temporary ordering mask in block 1150. The current ordering mask defines the relative transmission order of the received packets.

In block 1160, the received packet is checked whether it is older than previously received packet. If so, the received packet is discarded in block 1170. If not, the current ordering mask is marked as the most recently used mask in block 1180. The most recently used mask corresponds to the mask pointed to by ordering mask pointer 620 (FIG. 6).

The embodiment of FIG. 11 discards the received packet when it is older than a previously received packet. In another embodiment, block 1170 buffers received packets rather than discarding received packets, such that the correct order of received packets can be re-created. This can be accomplished using buffers such as error detection device and buffer 514 (FIG. 5), and error detection device and buffer 812 (FIG. 8).

FIG. 12 is a flowchart of a method for determining a packet order of a received packet in accordance with a preferred embodiment of the present invention. Method 1200 begins in block 1210 when at least one ordering mask is applied to a received packet. The at least one ordering mask is applied in a known order from a list of ordering masks to find a current ordering mask that was previously used to mask a received packet. In block

1220, ordering masks that are older than the current ordering mask are removed from the list of ordering masks.

5 In summary, the method and apparatus of the present invention provide a mechanism for masking packets prior to transmission and unmasking the packets after reception. While we have shown and described specific embodiments of the present invention, further modifications and improvements will occur to those skilled in the art. For example, while the method and apparatus of the present invention have been described with reference to discerning the correct order of transmission, the method and apparatus of the present invention also apply to other applications for packet  
10 identification. For example, in some embodiments of the present invention, the masking and unmasking are used for identifying different types of packets such as data packets and control packets. We desire it to be understood, therefore, that this invention is not limited to the particular forms shown, and  
15 we intend in the appended claims to cover all modifications that do not depart from the spirit and scope of this invention.